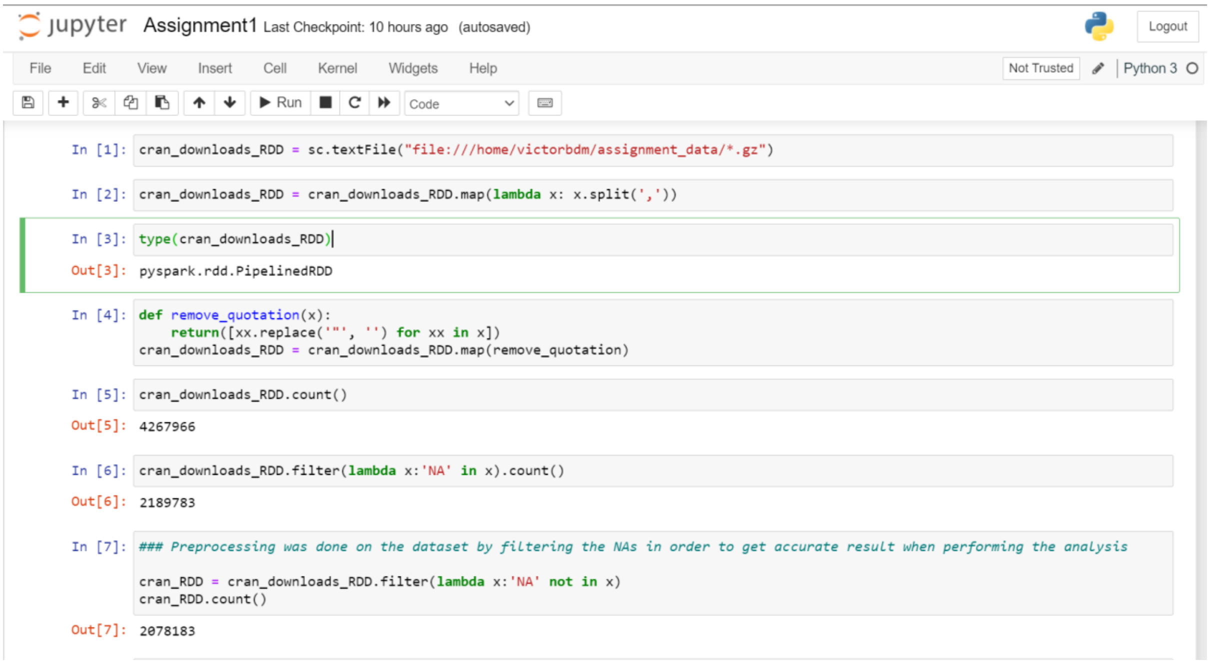
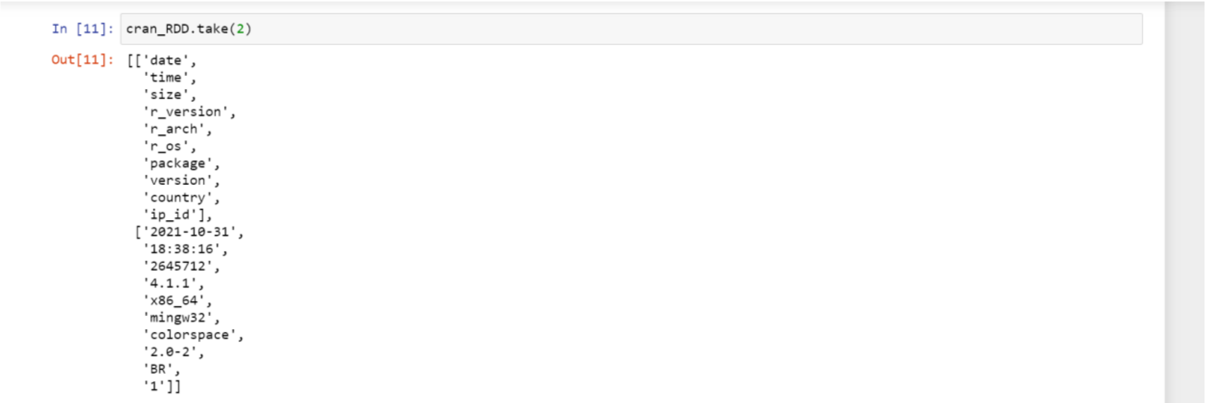
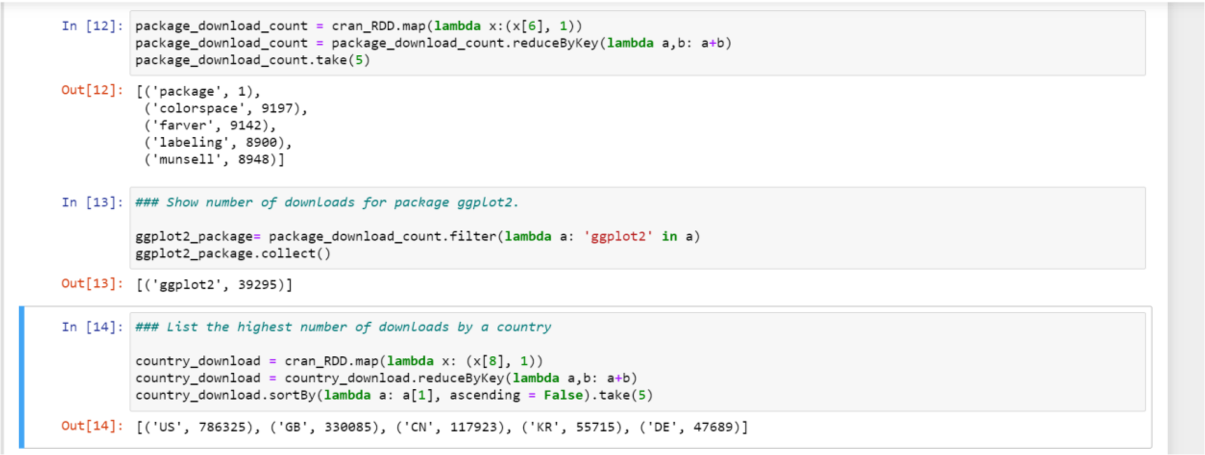
Victor Abiodun Adepoju -3023507



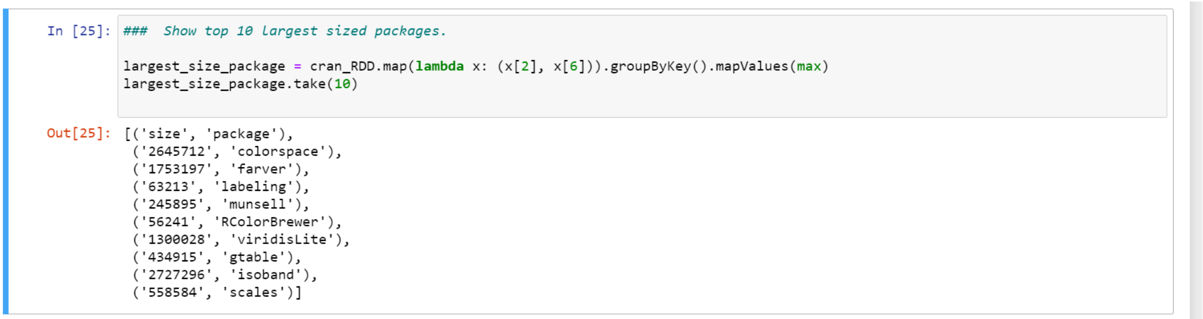
```
In [1]: cran_downloads_RDD = sc.textFile("file:///home/victorbdm/assignment_data/*.gz")

In [2]: cran_downloads_RDD = cran_downloads_RDD.map(lambda x: x.split(','))

In [3]: type(cran_downloads_RDD)

Out[3]: pyspark.rdd.PipelinedRDD

In [4]: def remove_quotation(x):
            return([xx.replace('"', '') for xx in x])
        cran_downloads_RDD = cran_downloads_RDD.map(remove_quotation)

In [5]: cran_downloads_RDD.count()

Out[5]: 4267966

In [6]: cran_downloads_RDD.filter(lambda x:'NA' in x).count()

Out[6]: 2189783

In [7]: ### Preprocessing was done on the dataset by filtering the NAs in order to get accurate result when performing the analysis

        cran_RDD = cran_downloads_RDD.filter(lambda x:'NA' not in x)
        cran_RDD.count()

Out[7]: 2078183
```



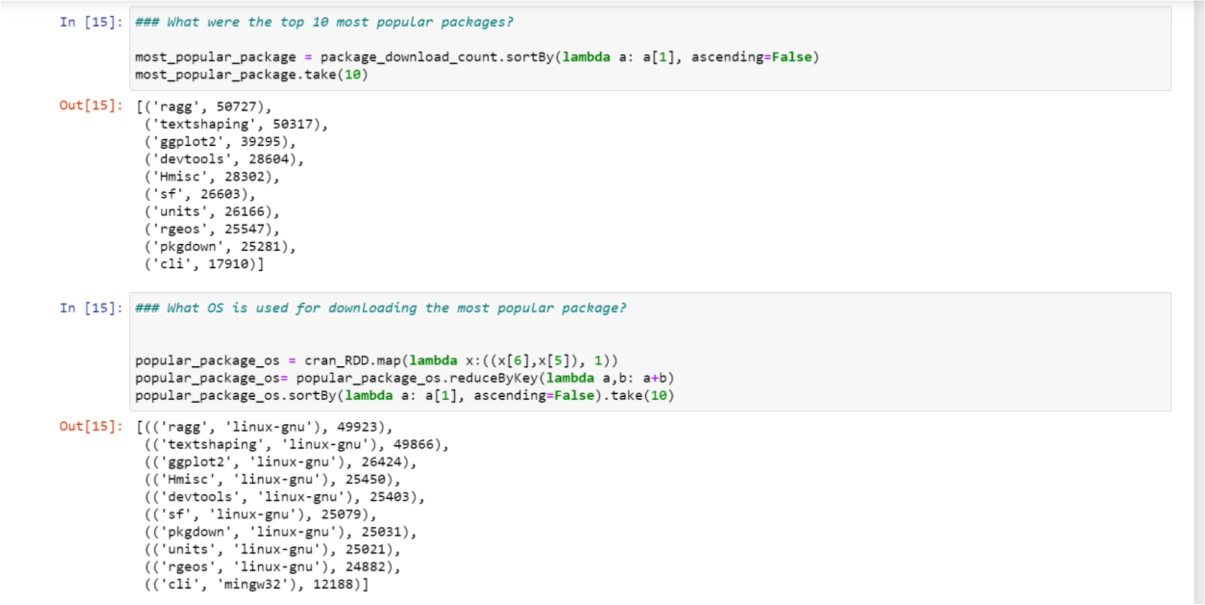
```
In [11]: cran_RDD.take(2)

Out[11]: [['date',
           'time',
           'size',
           'r_version',
           'r_arch',
           'r_os',
           'package',
           'version',
           'country',
           'ip_id'],
          ['2021-10-31',
           '18:38:16',
           '2645712',
           '4.1.1',
           'x86_64',
           'mingw32',
           'colorspace',
           '2.0-2',
           'BR',
           '1']]
```



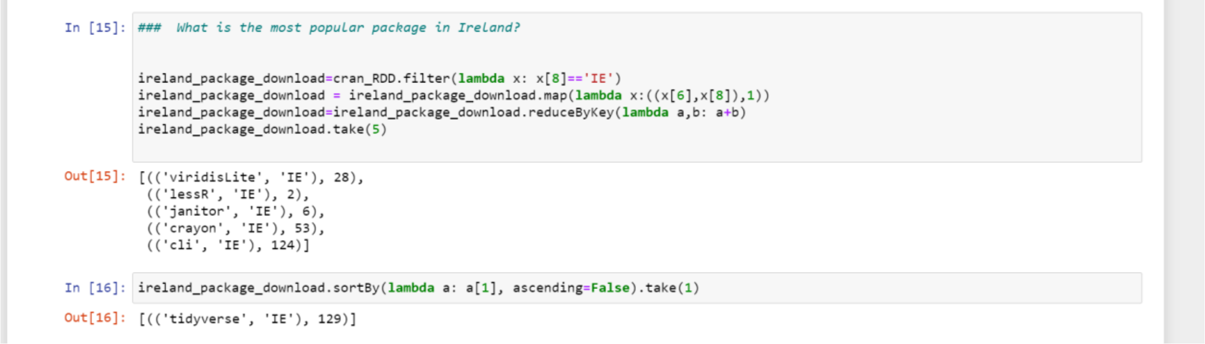
```
In [12]: package_download_count = cran_RDD.map(lambda x:(x[6], 1))
         package_download_count = package_download_count.reduceByKey(lambda a,b: a+b)
         package_download_count.take(5)

Out[12]: [('package', 1),
          ('colorspace', 9197),
          ('farver', 9142),
          ('labeling', 8900),
          ('munsell', 8948)]

In [13]: ### Show number of downloads for package ggplot2.

         ggplot2_package= package_download_count.filter(lambda a: 'ggplot2' in a)
         ggplot2_package.collect()

Out[13]: [('ggplot2', 39295)]

In [14]: ### List the highest number of downloads by a country

         country_download = cran_RDD.map(lambda x: (x[8], 1))
         country_download = country_download.reduceByKey(lambda a,b: a+b)
         country_download.sortBy(lambda a: a[1], ascending = False).take(5)

Out[14]: [('US', 786325), ('GB', 330085), ('CN', 117923), ('KR', 55715), ('DE', 47689)]
```



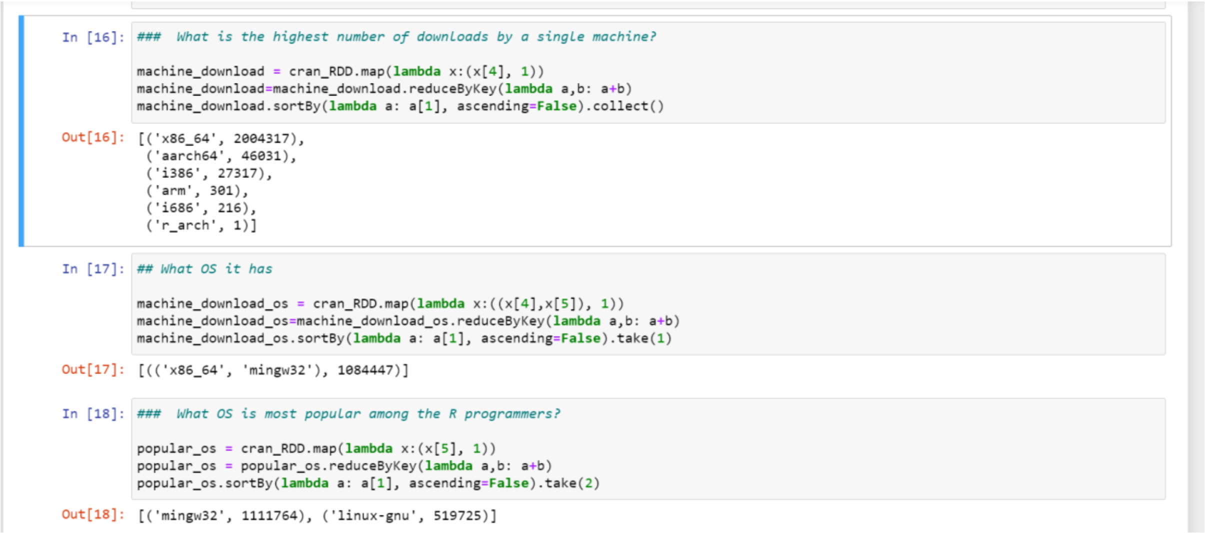
```
In [25]: ###  Show top 10 largest sized packages.

         largest_size_package = cran_RDD.map(lambda x: (x[2], x[6])).groupByKey().mapValues(max)
         largest_size_package.take(10)

Out[25]: [('size', 'package'),
          ('2645712', 'colorspace'),
          ('1753197', 'farver'),
          ('63213', 'labeling'),
          ('245895', 'munsell'),
          ('56241', 'RColorBrewer'),
          ('1300028', 'viridisLite'),
          ('434915', 'gtable'),
          ('2727296', 'isoband'),
          ('558584', 'scales')]
```



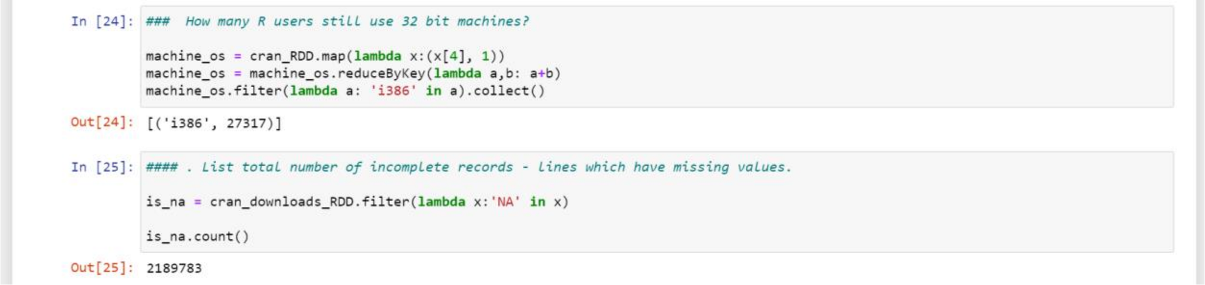
```
In [15]: ### What were the top 10 most popular packages?

         most_popular_package = package_download_count.sortBy(lambda a: a[1], ascending=False)
         most_popular_package.take(10)

Out[15]: [('ragg', 50727),
          ('textshaping', 50317),
          ('ggplot2', 39295),
          ('devtools', 28604),
          ('Hmisc', 28302),
          ('sf', 26603),
          ('units', 26166),
          ('rgeos', 25547),
          ('pkgdown', 25281),
          ('cli', 17910)]

In [15]: ### What OS is used for downloading the most popular package?


         popular_package_os = cran_RDD.map(lambda x:((x[6],x[5]), 1))
         popular_package_os= popular_package_os.reduceByKey(lambda a,b: a+b)
         popular_package_os.sortBy(lambda a: a[1], ascending=False).take(10)

Out[15]: [(('ragg', 'linux-gnu'), 49923),
          (('textshaping', 'linux-gnu'), 49866),
          (('ggplot2', 'linux-gnu'), 26424),
          (('Hmisc', 'linux-gnu'), 25450),
          (('devtools', 'linux-gnu'), 25403),
          (('sf', 'linux-gnu'), 25079),
          (('pkgdown', 'linux-gnu'), 25031),
          (('units', 'linux-gnu'), 25021),
          (('rgeos', 'linux-gnu'), 24882),
          (('cli', 'mingw32'), 12188)]
```



```
In [15]: ###  What is the most popular package in Ireland?


         ireland_package_download=cran_RDD.filter(lambda x: x[8]=='IE')
         ireland_package_download = ireland_package_download.map(lambda x:((x[6],x[8]),1))
         ireland_package_download=ireland_package_download.reduceByKey(lambda a,b: a+b)
         ireland_package_download.take(5)

Out[15]: [(('viridisLite', 'IE'), 28),
          (('lessR', 'IE'), 2),
          (('janitor', 'IE'), 6),
          (('crayon', 'IE'), 53),
          (('cli', 'IE'), 124)]

In [16]: ireland_package_download.sortBy(lambda a: a[1], ascending=False).take(1)

Out[16]: [(('tidyverse', 'IE'), 129)]
```



```
In [16]: ###  What is the highest number of downloads by a single machine?

         machine_download = cran_RDD.map(lambda x:(x[4], 1))
         machine_download=machine_download.reduceByKey(lambda a,b: a+b)
         machine_download.sortBy(lambda a: a[1], ascending=False).collect()

Out[16]: [('x86_64', 2004317),
          ('aarch64', 46031),
          ('i386', 27317),
          ('arm', 301),
          ('i686', 216),
          ('r_arch', 1)]

In [17]: ## What OS it has

         machine_download_os = cran_RDD.map(lambda x:((x[4],x[5]), 1))
         machine_download_os=machine_download_os.reduceByKey(lambda a,b: a+b)
         machine_download_os.sortBy(lambda a: a[1], ascending=False).take(1)

Out[17]: [(('x86_64', 'mingw32'), 1084447)]

In [18]: ###  What OS is most popular among the R programmers?

         popular_os = cran_RDD.map(lambda x:(x[5], 1))
         popular_os = popular_os.reduceByKey(lambda a,b: a+b)
         popular_os.sortBy(lambda a: a[1], ascending=False).take(2)

Out[18]: [('mingw32', 1111764), ('linux-gnu', 519725)]
```



```
In [24]: ###  How many R users still use 32 bit machines?

         machine_os = cran_RDD.map(lambda x:(x[4], 1))
         machine_os = machine_os.reduceByKey(lambda a,b: a+b)
         machine_os.filter(lambda a: 'i386' in a).collect()

Out[24]: [('i386', 27317)]

In [25]: #### . List total number of incomplete records - lines which have missing values.

         is_na = cran_downloads_RDD.filter(lambda x:'NA' in x)

         is_na.count()

Out[25]: 2189783
```